Task Scheduling in Energy Harvesting Real-time Embedded Systems

Maryline Chetto

Information Technology & Software Engineering

Chetto, J Inform Tech Softw Eng 2012, 2:3
http://dx.doi.org/10.4172/2165-7866.1000e106

Editorial Open Access

Task Scheduling in Energy Harvesting Real-time Embedded Systems

Maryline Chetto*

L'UNAM University–IRCCyN (Institut de Recherche en Communications et Cybernétique de Nantes), UMR CNRS 6597, Nantes, France

Abstract

Harvesting energy from the environment is very desirable for many emerging applications that use embedded devices. Energy harvesting also known as energy scavenging enables us to guarantee quasi-perpetual system operation for wireless sensors, medical implants, etc. without requiring human intervention which is normally necessary for recharging batteries in classical battery-operated systems. Nevertheless, energy harvesting calls for solving numerous technological problems in relation with chemistry when batteries are used for temporary energy storage for example, power management of the embedded computing system that consumes the energy, etc. And this latter problem becomes more complex when the embedded system has real-time constraints i.e. deadlines attached to computations. This paper surveys the main issues involved in designing energy harvesting embedded systems that present strict timing requirements.

Introduction to Energy Harvesting

Energy harvesting is a technique that generates electricity from renewable energy sources [1]. Although it has been used for a long time (windmill, solar panel, etc.), the novation lies in the design of smart embedded systems with efficient energy harvesting capabilities while satisfying reliability, dimensioning and timing requirements that characterize modern embedded computers. Any energy harvesting system is composed of three parts which respectively concern conversion from one form of energy in electricity, storage of the energy once it has been harvested from the source and finally consumption of the energy by the embedded computing system. Many forms of energy can be harvested from the environment and consequently they use different types of generators. They include mechanical, thermal, photovoltaic, electromagnetic, biological, and chemical energy. Solar energy and mechanical energy are certainly the most prevalent ones. For example, the mechanical energy can be drained from ambient vibration.

Consuming the energy only at the time of its production is not always possible. Consequently, an energy storage unit permits the embedded system to continue operation even when there is no energy to harvest. Generally, rechargeable batteries are used for long term storage. In addition, supercapacitors or ultracapacitors are commonly used for storing transiently the energy from several seconds to minutes without involving aging and rate-capacity problems.

Power Management and Task Scheduling Issues

Energy harvesting embedded systems have received attention from the research community from about one decade. The main issue in designing an energy harvesting embedded system -besides efficiently extracting and storing the energy- is to dynamically adapt the activity of the embedded system to the available energy. In other terms, we have to implement a power management policy that is permanently aware of the operating conditions mainly given by the level of energy currently stored in the storage unit (battery or supercapacitor) and the future incoming energy.

The conventional objective of the power management policy in a battery-powered system such as a notebook computer or a cellular phone is to maximize its lifetime under a total energy constraint. There, the focus is on the reduction of the power consumption of the device with Dynamic Power Management (DPM) or Dynamic Voltage and Frequency Scaling (DVFS) [2]. DPM and DVFS techniques reveal to be effective in reducing the energy dissipation but they do not prevent any device to exhaust the battery. In energy harvesting systems, the challenging issue is to operate in an energy neutral mode, consuming only as much energy as harvested. Consequently, this leads to the possibility of indefinitely long lifetime if limitation due to hardware longevity is forgotten. Nevertheless, the objective of energy neutrality can be attained only if certain operating conditions are satisfied, in particular related to the average power generated by the harvesting device and the capacity of the energy storage unit. How to perform power management and how to schedule real-time tasks energy-efficiently are consequently the main issues of research in the domain of energy harvesting real-time embedded systems.

Real-time embedded systems are characterized by the need for running multiple tasks (programs) on a processing unit such as a micro-controller. Scheduling these tasks on that processor so that real-time constraints are met is a difficult problem [3]. The scheduling problem consists of deciding the order, the start time and the running time of tasks with known characteristics such as deadline, periodicity, duration and energy consumption. We generally classify scheduling policies for real-time systems as static ones when tasks execute in a fixed order determined offline or dynamic ones when the order of execution is decided online. A dynamic scheduling policy is very often based on priority where priority intuitively represents a measure of the urgency of each task. Earliest Deadline First (dynamic priority depending on urgency) and Rate Monotonic (fixed priority depending on period) can support sophisticated task set characteristics such as deadlines, precedence constraints, shared resources, jitter, etc., [4]. Despite the significant body of results in real-time scheduling, many real world problems are not easily supported, including energy harvesting.

State of the Art

Developing both good power management techniques and good task scheduling algorithms dedicated to energy harvesting systems are two current challenges. One decade ago, researchers started to address

***Corresponding author:** Maryline Chetto, L'UNAM University–IRCCyN (Institut de Recherche en Communications et Cybernétique de Nantes), UMR CNRS 6597, Nantes, France, E-mail: maryline.chetto@univ-nantes.fr

Received September 13, 2012; **Accepted** September 14, 2012; **Published** September 17, 2012

Citation: Chetto M (2012) Task Scheduling in Energy Harvesting Real-time Embedded Systems. J Inform Tech Softw Eng 2:e106. doi:10.4172/2165-7866.1000e106

power and scheduling issues with the objective of either minimizing power usage under timing constraints or maximizing the system performance under the energy constraints. But in most of works, they do not consider the rechargeability of the batteries. For example, EDF (Earliest Deadline First) and RM (Rate Monotonic) scheduling have been extended to variable-voltage processors. The idea is to save power by slowing down the processor just enough to meet the deadlines [5]. But solely applying these techniques has limitations in energy harvesting systems because they minimize the processing power, rather than they dynamically manage power according to the profiles of both available energy and processor workload.

In [6], C. Moser et al. propose a real-time scheduling algorithm, called Lazy Scheduling Algorithm (LSA). LSA is energy-clairvoyant, i.e. the generated energy in the future is perfectly known. An optimal task ordering can be determined based on the prediction of the available energy in the future by taking into account available time as well as processable energy. In that work, the real-time system is built around a mono-processor architecture that draws the energy from storage and uses it to process tasks (periodic or non periodic ones) with arrival time, deadline, and worst case execution time. The worst case execution time corresponds to the maximum energy demand of the task. Tasks are preemptable and executed according to the EDF policy. The energy storage unit has a precisely known capacity. The stored energy is assumed to be known at any time instant and is no more than the storage capacity. If the stored energy reaches the capacity, the incoming harvested energy overflows the storage and is discarded. According to LSA, the processor executes all tasks at full power; and the system starts executing a task if the task is ready and has the earliest deadline among all ready tasks and the system is able to keep on running at the maximum power until the deadline of the task.

In contrast to greedy scheduling algorithms such as EDF, LSA hesitates to power tasks until it is necessary to respect timing constraints. In that sense, tasks are executed neither as soon as possible nor as late as possible. So the purpose of LSA is to find the optimal start time to begin execution of every task. The authors of LSA also describe an admittance test that decides whether a set of tasks can be scheduled without violating deadlines. Another crucial question which has been solved is how to dimension the capacity of the battery that ensures continuous operation. The performance evaluation study demonstrates that achievable capacity savings between 20% and 45% are obtained compared with the classical EDF policy. The main drawback of the LSA algorithm is that a task is executed at the maximum power and may be finished well before its deadline. In this case, the task slack would be wasted; more importantly, the limited energy is unnecessarily squandered. As a consequence, future tasks have to violate their deadlines because of energy shortage.

In [7], an energy-aware dynamic voltage and frequency selection algorithm is proposed to enhance the performance of LSA. The so-called EA-DVFS algorithm adjusts the processor's behavior depending on the sum of the stored energy and the harvested energy in a future duration. Specifically, if the system has sufficient energy, tasks are executed at full speed; otherwise, the processor slows down task execution to save energy. EA-DVFS saves energy by slowing down the task executing when the system is in shortage of energy. It improves performance in terms of remaining energy, deadline miss rate and minimum storage capacity for maintaining zero deadline miss rate. Additional works permit to achieve substantive energy savings with some modifications in EA-DVFS.

In [8], it is shown how to dynamically manage power in a single processor system where tasks consume energy with time varying consumption power. In a real application, instantaneous power consumed by tasks may vary along time depending on circuitry and devices required by the tasks. Consequently the worst case energy requirement and the worst case execution time of a task are completely independent variables. We present an on-line scheduling scheme called EDeg (Earliest Deadline with energy guarantee) where each task has a WCET (Worst Case Execution Time) and a WCEC (Worst Case Enercy Consumption). EDeg is a variation of EDF that jointly handles constraints from the energy domain and from the time domain. It is based on two fundamental concepts, namely slack time and slack energy. This guarantees the absence of energy starvation for every task as long as the application is feasible. The intuition behind EDeg is to run tasks as long as the energy storage is sufficient to provide energy for all future occurring tasks, considering their timing and energy requirements and the replenishment rate of the storage unit. When this condition is not verified, the processor has to idle so that the storage unit recharges as much as possible and as long as the system will be able to meet all the deadlines.

Conclusion

Energy harvesting technology has been explored very recently in the aim of bringing solutions to the energy problem and extending the embedded system operating duration. As an energy harvesting system draws parts or all of its operating energy from its ambient energy sources, it has potential to overcome the energy constraint imposed by traditional battery-powered embedded systems. New applications of energy harvesting technology for embedded systems are beginning to drive economic development in many sectors. It concerns as well the high technology sectors as the general public products in which wireless sensor networks are used in a variety of embedded applications, such as environmental applications and military applications. This paper has provided a short and non exhaustive overview on power management and task scheduling in energy harvesting embedded systems. To take full advantage of the energy harvesting technology, efficient scheduling algorithms must consider the finite capacity for energy storage and use prediction on the energy harvesting income in order to adapt at run-time the executing set of prioritized tasks. While there has been extensive research on classical real-time scheduling, those specific to energy harvesting systems are just emerging.

References

1. Priya S, Inman DJ (2008) Energy Harvesting Technologies. Springer 544.
2. Quan G, Hu XS (2007) Designing Embedded Processors, A Low Power Perspective. Springer Netherlands Chapter 10.
3. Liu J W S (2000) Real-Time Systems. Prentice Hall.
4. Liu CL, Layland JW (1973) Scheduling algorithms for multiprogramming in a hard real time environment. Journal of the ACM 20: 46–61.
5. Qian D, Zhang Z, Tian X, Hu C (2010) Low power scheduling for periodic real-time systems with Dynamic Voltage Scaling processor. International Conference on Computer Application and System Modeling (ICCASM) 11: 244-248.
6. Moser C, Brunelli D, Thiele L, Benini L (2007) Real-time scheduling for energy harvesting sensor nodes. Real-Time Systems 37: 233-260.
7. Liu S, Qiu Q, Wu Q (2008) Energy aware dynamic voltage and frequency selection for real-time systems with energy harvesting. Proceedings of the conference on Design, automation and test in Europe 236-241.
8. El Ghor H, Chetto M, Chehade RH (2011) A real-time scheduling framework for embedded systems with environmental energy harvesting. Computers and Electrical Engineering 37: 498-510.